Kristianstad University
www.hkr.se

Fredrik Frisk
Computer Science Department

Content and purpose of this lab

During the lab we will start acquiring some data for late machine learning experiments. some machine learning experiments using your own recorded data. We will work with our own recorded data. We will start with a smaller recording to just get used to the needed software.

We will use pandas in this lab, mostly due to the plotting functions and the handling of csv-files.

Part one will focus on record data from sensors, import the data as pandas objects, actually a dataframe. We will also plot them and do some preprocessing.

Save the code you are writing in this lab for future use. To pass the lab you need to solve/program the different bullet points and be able to explain your results. If you are not finished with the all the bullet points the remaining ones are a part of the required preparations for part 2 of the labs. The lab report in the end is an individual report, but you are allowed to work two and two with one exception all of you must record your own sensor data.

Required preparations

The following you need have done before the lab:

- Record all the data discussed in the chapter “Acquire some data”
- Be prepared to show the csv-files

If you have not done all the recordings and fail to show the files when the lab starts <u>you have to leave</u> and you are welcome back to the lab when the reexamination takes place.

Acquire some data

We will start in an easy pace. Record accelerometer data and rategyro data, all three directions, for two different positions and one type of movement,

- standing
- laying down.
- Walking

Put the phone in your pocket while you do the recording. *Note, keep the phone in the same position for all recordings!* Otherwise it will be hard to do some machine learning. Transfer the data so it is accessible in your python environment. Each recording should be at least 15 s. Follow the following procedure:

1. Start the recording
2. Start walking, stand or lay down.
3. Stop what you do
4. Stop the recording

Below you will find the overview of all recordings needed

Position/class	Duration	# of recordings	Sensors	Sample Frequency
Laying down	15 s	3	Accelerometer + Rategyro	100 Hz
Standing	15 s	3	Accelerometer + Rategyro	100 Hz
Walking	15 s	3	Accelerometer + Rategyro	100 Hz

Now you have 3 x 3 recording, that is all in all 9 recordings. On some phones the accelerometer data and rategyro data are stored in different files, so in that case you have 18 files.

Other Preparations (These will save you time)

Work through the tools_pandas.ipynb notebook. You need to have an understanding of the following items before the lab:

- Panda Series
- Panda Dataframe
- .loc
- .iloc
- .plot
- Adding and removing columns

That means that you must know what these items are and how to use them (in some respects). This notebook also refers to dictionaries. You need to know this python object as well, some information you can find here:

https://www.w3schools.com/python/python_dictionaries.asp

There is a lot of other stuff as well in this notebook, but the bullet points above I suggest you start with.

Data analysis (Here the lab starts!)

Each bullet point means a task that you need to show to pass the lab. Make a heading for each group of bullet points, put the necessary code in one or more cells. Answer the questions in separate markdown cells. Note it is a good idea to copy the question so it is easy to se what you are answering.

Needed imports:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

The dataset is imported using the panda method shown below , or similar:

```
dataset_acc = pd.read_csv(filename, usecols=[1,2,3],
names=['ax', 'ay', 'az'], skiprows = [0])
```

The import of the rategyro data is similar, just change the names to `gx, gy`, and `gz`. Some of the smartphones actually records five columns, you then need to use the column 2,3, and 4.

- Plot the accelerometer values using the the plot functions accessible for panda dataframes. Plot all accelerometer values, i.e. all samples for all components of the accelerometer vector.
- What are we measuring with the accelerometer?

As you may have noticed the scale on the x-axis is the index of the rows in the dataframe.

- By looking at the plots can you see the difference between the the three classes? Explain, and take some notes for future work.

- Work through the above four bullet points for the rategyro values as well.

You can easily plot a dataframe by using the method .plot()

Preprocessing the data

Once again plot the data for each recording. You should see some irregular data in the beginning of the recording and in the end of the recording.

- Why do you have these irregularities in your recordings? Or maybe you do not, how come?

- Create a python function that can read one accelerometer file and one rategyro file, remove a specific number of samples in the beginning and in the end and output a dataframe with six columns, that is all the accelerometer and ratygyro attributes. What do you need as input to the function?

Now you have a "cleaned" dataframe for each recording.

- Store each dataframe, one for each recording, in a binary file using the to_pickle function.

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_pickle.html

Note. choose a clever name on the pickle file so easily now what it is. One suggestion is classname and a number for example `walk_1`.

Python

Divide the data into two sets, training set and test set. Store 2 of 3 files in the training folder and 1 of 3 files in the test folder. Note: it is the binary pickle files we talk about.

- Create a python function that can read all binary-files from one class. The function should return a dataframe *x* with all the data and also a column with information of which class the data belongs to. The dataframe has now 7 columns.

You need the `.concat` dataframe methods. Some examples you can find below:

https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html

https://pandas.pydata.org/docs/reference/api/pandas.concat.htm

Set ignore_index to True, to avoid difficulties regarding merging the dataframes.